
Programmeerimise põhikursus

ITI0010

- Ülevaade kursuse eesmärkidest
 - Kirjandus, viited
 - Toimumisajad, lektor, assistendid, eksam jms
 - Praktikumi tööd, tähtajad, põhimõtted
 - Loengukava
-
- Java mootor, kompileerimine, eripärad

Kursuse eesmärk, eeldused, põhimõtted

- Õpetada programmeerimise aluseid ja tehnoloogiat üldiselt
 - Eeldades kergelt varasemat kokkupuutumist programmeerimisega vähemalt eeldusaine Informaatika II mahus.
- Rõhk on praktiliste kogemuste ja oskuste omandamisel: **praktikumitöö on tähtsam, kui loeng!**
- Baaskeelena kasutatakse **Javat**, vältides samas eriliselt Java-päraseid meetodeid, teeke jms. Ei ole spetsiaalne “Java-kursus”.
- Praktikumides loodavad rakendused on reeglina käsurea ja omaette akendega rakendused, mitte apletid.
- Kursuse lõpetaja on alles **ALGAJA** programmeerija!

- Loengud: reedeti kell 12.00-13.30 ruumis VII-131 (kohe fuajee kõrval).
- Loenguid loeb Aivo Anier
- Kokku 16 (?) loengut
- Praktikumid eraldi AK arvutiklassides: eri gruppidel eri aegadel, 1 praktikum nädalas
- Praktikume korraldavad assistendid.
 - Ivor Lõõbas
 - Ago Luberg
- Kursus lõpeb **EKSAMIGA**.
- Kursuse edukaks lõpetamiseks tuleb:
 - Teha ära praktilised tööd, neid tuleb 4 (40p)
 - Sooritada edukalt eksam (60p)

Eksami ja praktikumi ülesanded ja nõuded

- **Eksam:** kirjalik, lisamaterjale kasutada ei või
 - **Praktikumid:**
 - Iseseisev töö (ei mingeid gruppe)
 - Sissejuhatav ülesanne: see nädal
 - Neli ülesannet:
 - Esimene rakendus: tikumäng, ainult tekstiliides.
- Tähtaeg: 28. september.**

Need on esialgsed variandid, vt. eelmise aasta lehte:

- Tekstifailidest mikroandmebaas, ainult tekstiliides.
Tähtaeg: 26 oktoober.
- GUI-ga võrgurakendus (**http**-d kasutav klient).
Tähtaeg 23. november
- GUI-ga connect-four mäng (**otsing** mängupuust).
Tähtaeg: 28. detsember

Olulist praktikumi kohta

- Eksamile pääsemiseks peavad olema sooritatud **vähemalt kolm** neljast praktikumi-ülesandest.
 - Juhul, kui üks neljast ülesandest jääb realiseerimata, kaotate 10p.
 - 51-60, 61-70, ..., 91-100
- Iga **hilinenud ülesande** punktid jagatakse kahega.
- Pärast 18. detsembrit ei aktsepteerita enam ühtegi ülesannet.
- Iga ülesanne tuleb programmeerida **iseseisvalt** ja näidata praktikumi juhendajale praktikumi ajal ette.
 - E-postiga saadetud lahendusi **ei aktsepteerita!**
- NB! Ülesande arvestamiseks ütle juhendajale praktikumi ajal ise, kui soovid lahendatud ülesannet näidata: juhendaja ei hakka reeglina ise õigeagselt uurima, et kas on ülesanne ikka tehtud.

- Kursuse www-leht:

http://www.lambda.ee/index/Programmeerimise_põhikursus

- Peamine õpik kursuse juurde on tasuta e-book:

David Eck, “Introduction to Programming Using Java”

Fifth Edition, 2006

<http://math.hws.edu/javanotes/>

- Materjalid ilmuvad loengute eel või järel võrku. Alati on olemas loengu üldplaan ja põhipunktid, kuid palju seletavaid detaile on võrgus puudu.

Eksamiküsimuse näide

```
public class PrintThread1 extends Thread {
    String name;
    public static void main(String args[]) {
        PrintThread1 a = new PrintThread1("*");
        PrintThread1 b = new PrintThread1("-");
        a.start();
        b.start();
    }
    public PrintThread1(String name) {
        this.name = name;
    }
    public void run() {
        for (int i=1; i<4 ; i++) System.out.print(i+": "+name);
    }
}
```


?

Kursuse teemad

- September: juhtimiskonstruktsioonid, andmed
- Oktoober: objektid, kasutajaliides ja graafika
- November: otsing, rekursioon, puud
- Detsember: paralleeltöö, metoodika, (varu)

Kursuse esialgne loengukava

1. Sissejuhatus, meeldetuletus, Java kasutamine, Java ja C võrdlus, lihtsad näited.
2. Juhtimiskonstruktsioonid ja tsüklid. Funktsioonid ehk alamprogrammid. Lühisissejuhatus klassidesse..
3. Andmed. Massiivid ja struktuurid. Failid. Võrk. Veatöötlus.
4. Failide kasutamise jätk. Binaarne otsing, sorteerimine, rekursioon.
5. Sorteerimise jätk. Seejärel: objektid ja klassid.
6. Objektide jätk. Seejärel: Graafika ja kasutajaliides.
7. Kasutajaliides: komponendid, sündmused.
8. Kolmas praktikum: olulist, näiteid võrgurakendusest.
9. Kolmas praktikum: olulist, näiteid GUI tegemisest.
10. GUI tegemise lisainfot. Otsingualgoritmide algus.
11. Otsingualgoritmide jätk. Neljanda praktikumi sissejuhatus.
12. Keerukamad andmestruktuurid. Puud, hash.
13. Avaldiste süntaksianalüüs.
14. Serverid, threadid ja protsessid: paralleeltöö.
15. Arendusmetoodikad ja abivahendid.
16. KORDAMINE JA RESERV
17. KORDAMINE JA RESERV

Vajalik tarkvara kodus töötamiseks (kes soovib)

- Java versioon J2SE (Java 2 standard edition) JDK (software development kit).
 - Tõmba SUN-i lehelt <http://java.sun.com/j2se/downloads.html> vähemalt 1.5 või uuem versioon.
 - Veidi vanemate versioonide (1.3.1 ja uuemad) kasutamine on sama OK.
 - NB! Vajalik on JDK (vana nimi oli SDK), mitte JRE, ja nimelt J2SE (standard edition)
 - mitte J2ME (micro edition) või J2EE (enterprise edition).
- Hea ja lihtne programmeerisele kohandatud tekstiredaktor. Soovitan redaktorit **SciTE**: <http://www.scintilla.org/SciTE.html>
- **Eclipse** IDE for Java Developers, www.eclipse.org
- Käsurida

Programmi kirjutamise etapid:

- getting the program text into the computer,
- compiling the program, and
- running the compiled program.

Final step - running the program - either as

- Application - program running without a www browser
- Applet- program running in a www browser
- Servlet- program running in a (web) server

Java: APPLICATION

```
public class HelloWorld {  
  
    // A program to display the message  
    // "Hello World!" on standard output  
  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
  
}    // end of class HelloWorld
```

- kompileerime: **javac HelloWorld.java => HelloWorld.class**
- paneme käsurealt käima: **java HelloWorld**

function called **main**, with a definition:

```
public static void main(String[] args) {  
    statements  
}
```

Java: APPLET

```
import java.applet.*;
import java.awt.* ;

public class Helloa extends Applet
{
    public void init() {
        resize(300, 500);
    }

    public void paint(Graphics g) {
        g.drawString("Hello", 10,50);
    }
}
```

kompileerime: javac Helloa.java => Helloa.class
HTML failis: Tere.html

See on Hello applet:

```
<applet code = "Helloa" width=120 height=120>
</applet>
```

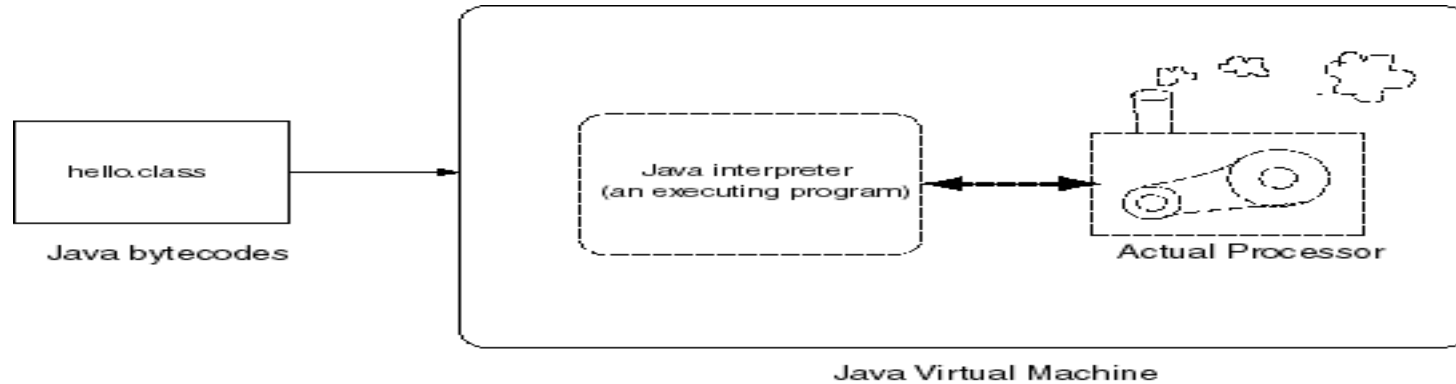
Hello applet loppes.

Compiler, bytecodes



Java Program Translation

Executing bytecodes



Java Bytecode Interpretation on a Virtual Machine